FAHES: Detecting Disguised Missing Values

Abdulhakim A. Qahtan, Ahmed Elmagarmid, Mourad Ouzzani, Nan Tang
Qatar Computing Research Institute, HBKU, Qatar
{*aqahtan, aelmagarmid, mouzzani, ntang*}*@hbku.edu.qa*

***Abstract*—It is well established that missing values, if not dealt with properly, may lead to poor data analytics models, misleading conclusions, and limitation in the generalization of findings. A key challenge in detecting these missing values is when they manifest themselves in a form that is otherwise valid, making it hard to distinguish them from other legitimate values. We propose to demonstrate FAHES, a system for detecting different types of disguised missing values (DMVs) which often occur in real world data. FAHES consists of several components, namely a profiler to generate rules for detecting repeated patterns, an outlier detection module, and a module to detect values that are used repeatedly in random records. Using several real world datasets, we will demonstrate how FAHES can easily catch DMVs.**

I. Introduction

Conducting data analytics and building data mining models on data with missing values is a challenging and well-recognized problem [1]. The performance of the models that are built using the data with missing values may be affected significantly as the missing values could introduce bias in the model. Also, the missing values could lead to misleading conclusions drawn from research studies or limit the generalization of the research findings [2].

Generally speaking, there are two types of missing values: explicit, *e.g.,* such as NULL values, and implicit, *a.k.a.* disguised missing values [1] (DMVs), *e.g.,* 11111111 for a phone number. Disguised missing values are usually caused by various reasons including: (i) the data field might not be applicable for certain records, *e.g.,* the number of children attribute in a table for records with value "single" in the marital status attribute; (ii) the person who enters the data might not care about providing the correct value or does not want to provide that information, which is common in *e.g.,* survey forms; (iii) the data value is not available at the time of entry, *e.g.,* a person would like to create a record in a hospital before receiving his insurance policy number; (iv) the correct value might not fit the data entry's checks set by the application so the user would enter a fake value that would allow him to complete an application form or get the record accepted by the DBMS, *e.g.,* a child is trying to create an email account but the system does not accept to create an account for under age children. Other reasons may arise in different applications.

While detecting DMVs that are coded using default values (*e.g.,* 00-00-0000 for date) is straightforward, detecting more general DMVs is challenging for several reasons: (a) DMVs in one table could represent valid values in other tables (no general rule could be applied for all datasets); (b) when the number of tables is large, using manual detection becomes impractical; (c) different persons and organization use different representations for the missing data (no global representation); and (d) faking the missing values with valid values becomes harder to detect, *e.g.,* entering the date-of-birth as 01/01/2000.

Table I
Examples of DMVs found in different data repositories.

Source	**Table Name**	**Column Name**	**DMV**
UCI ML	Pima Indians Diabetes	Diastolic Blood Pressurs	0
	adult	workclass	?
		eduction	Some College
U.S. FDA	Adverse Event Reporting System (AERS)	EVENT_DT	20010101, 20030101
data.gov	Allegheny County WIC Vendor Locatio	Ref_ID	-1
data.gov	Graduation Outcomes - School Level - Classes of 2005 - 2011 - SWD	Advanced Regents Num	s, -
data.gov.uk	Accidents 2015	Junction Control	-1

We have implemented FAHES[1] for DMVs detection. FAHES contains multiple components to detect different types of DMVs. The first component is a profiler that collects statistics about the data and generates a set of rules to detect data values with repeated patterns that do not fit in the data. The second component uses an outlier detector [3] to detect the extreme values that are used as DMVs. The third component handles the case of DMVs that are used repeatedly in random records. That is, when removing a value, the resulting empty cells follow a missing-at-random model. We use a new efficient implementation of DiMaC [4] to detect this type of DMVs.

To better grasp the severity and wide-spread use of DMVs, we looked at several real world data sets from different domains such as governments, education, environment and finance. For example, by looking at data.gov [5], a key resource of US government data with 200k data sets, we applied FAHES and confirmed manually the reported DMVs from (50+) tables. We also used FAHES on other data sets from data.gov.uk, FDA and UCI ML repository. We found that more than 90% of these tables have DMVs. Examples of what we have observed are shown in Table I,

[1] From the Arabic word that means inspector.

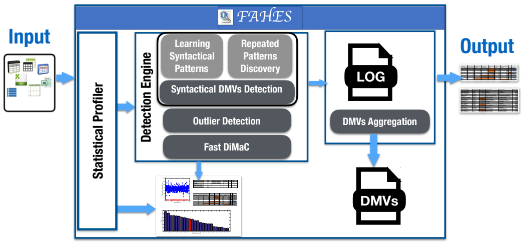


Figure 1. FAHES Architecture.

which include using single character to represent missing numerical values, zero for the area of a building, negative values in attributes that should take positive values and a phone number value within the records of many employees from different departments. Many of the discovered DMVs were hard to detect without the help of FAHES. This will be the first demo of a system that can detect different types of DMVs over a large number of datasets.

II. System Architecture

Let's start by discussing the characteristics of DMVs. Generally speaking, there are only a few distinct values that are used to replace the missing values in each attribute. These distinct values are used frequently within the data set most of the time. FAHES focuses on detecting the frequent DMVs since infrequent DMVs have less or no impact on the data analytics process. Figure 1 shows the architecture of FAHES which contains the following components.

A. Statistical Profiler

The data profiler collects statistics about the data that can be used by the detection engine. We collect two categories of statistics: (1) per table such as the number of records and the number of attributes and (2) per column such as the number of empty (null) cells and the number of numerical entries. For numerical values, we also count the number of positive/negative entries. These statistics helps the user understand the shape of the data so she can confirm if the reported DMVs are correct or not. They are also used to find the top frequent values that are more likely to be DMVs.

B. Detection Engine

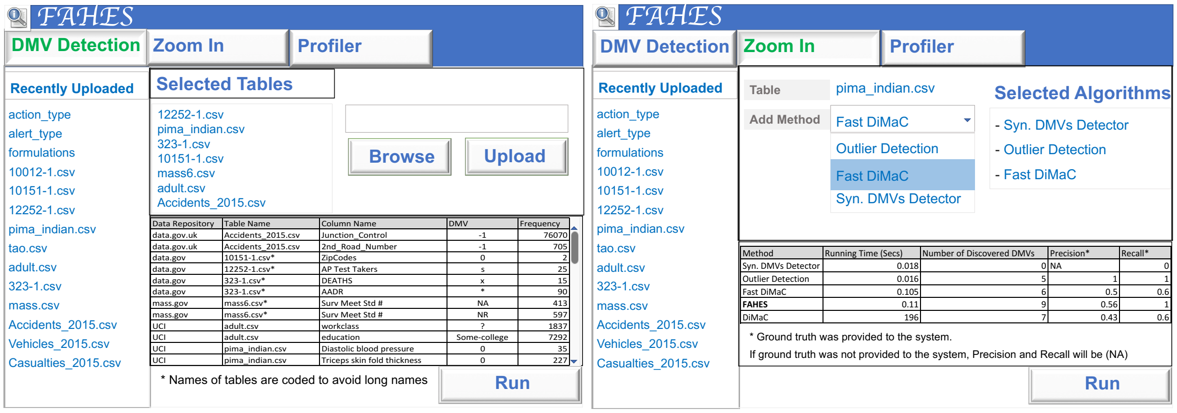
The detection engine include three main modules. Each module allows for detecting a specific type of DMVs.

(1) Syntactical DMVs Detector. The syntactical DMVs detector starts by building a structural description of the values in a given attribute. The structural description contains patters in the form similar to the regular expressions in the formal languages. The detector then discovers the set of patterns that represents the majority of the values in the given attribute. The values that have non-confirming syntactical structures (*i.e.,* cannot be generated by one of the dominating patterns) are likely to be erroneous. When these erroneous values belong to the top-K frequent values within that attribute, they are most likely DMVs. These values include single repeated string within an attribute that takes numerical data or a single string of special characters in an attribute with alphabetical strings, *e.g.,* in data.gov [5] many attributes include values such as (*, s, x, - and ?) that replaces the missing strings. Another example in road safety data sets from data.gov.uk, the value (-1) is used to replace the missing street numbers. It is unusual to have in an attribute a single repeated negative value with many distinct positive values unless that value is a DMV.

DMVs with repeated pattern such as a phone number attribute in a table with values 1111111111, 1212121212 or 1231231231 represent a special case of the syntactical DMVs; they might not be detected directly using the syntactical patterns of the attribute. We utilize hidden time patterns discovery algorithm [6] to detect data values with unusual patterns in the data. Applying the time pattern discovery algorithm on the set of distinct values in each attribute returns the values with repeated patterns. If there are a few values with repeated patterns that are used frequently within a large set of values with no patterns, then the confidence that these values are DMVs will be high.

(2) Outlier Detection for Detecting DMVs. In many cases, the data entry personnel uses values out of the range of the attribute values to replace (fill) the missing cells. For example, in the Pima Indians Diabetes data set [7], many values in the attribute diastolic blood pressure (DIA) were replaced by 0. In other data sets from traffic departments in UK (available at data.gov.uk), we noticed that unknown streets numbers were replaced with out of range negative values. We implemented a modified version of the outlier detector proposed in [3], where all the values that equal the value under test are removed from the data set before applying the outlierness test. That is because the DMVs are used frequently. Hence, including the other values will increase the probability density function at that sample, which will fool the detector in [3] so the value will not be reported as an outlier. Please note that neither all outliers are DMVs nor all DMVs are outliers. There is intersection between the two sets when values out of the range are used to represent the missing values.

(3) Detecting Missing-at-Random DMVs. Many DMVs follow a missing-at-random (MAR) model; the missing data depends on other observed data values [1]. The MAR model is a relaxed model of the missing-completely-at-random model (MCAR) [8]. In the MCAR, the missing values are randomly distributed across all records. In the MAR model, the missing values are randomly distributed within one or more sub-samples of the records. DMVs that follow MAR model have been studied in [9] and a tool called DiMaC has been developed and demonstrated in SIGKDD'08 [4]. .DiMaC claims that, if we remove the values that replace



Selected Tables: 12252-1.csv, pima_indian.csv, 323-1.csv, 10151-1.csv, mass6.csv, adult.csv, Accidents_2015.csv

Data Repository	Table Name	Column Name	DMV	Frequency
data.gov.uk	Accidents_2015.csv	Junction_Control	-1	76070
data.gov.uk	Accidents_2015.csv	2nd_Road_Number	-1	705
data.gov	10151-1.csv*	ZipCodes	0	2
data.gov	12252-1.csv*	AP Test Takers	s	25
data.gov	323-1.csv*	DEATHS	x	15
data.gov	323-1.csv*	AADR	*	90
mass.gov	mass6.csv*	Surv Meet Std #	NA	413
mass.gov	mass6.csv*	Surv Meet Std #	NR	597
UCI	adult.csv	workclass	?	1837
UCI	adult.csv	education	Some-college	7292
UCI	pima_indian.csv	Diastolic blood pressure	0	35
UCI	pima_indian.csv	Triceps skin fold thickness	0	227

* Names of tables are coded to avoid long names

(a)

Table: pima_indian.csv

Add Method: Fast DiMaC

Selected Algorithms

- Syn. DMVs Detector
- Outlier Detection
- Fast DiMaC

Method	Running Time (Secs)	Number of Discovered DMVs	Precision*	Recall*
Syn. DMVs Detector	0.018	0	NA	0
Outlier Detection	0.016	5	1	1
Fast DiMaC	0.105	6	0.5	0.6
FAHES	0.11	9	0.56	1
DiMaC	196	7	0.43	0.6

* Ground truth was provided to the system.

If ground truth was not provided to the system, Precision and Recall will be (NA)

(b)

Figure 2. Detecting the DMVs in FAHES.

the actual values in an attribute then the set of the resulting empty cells will follow a MAR model.

Based on the assumption that DMVs belong to a MAR model, a value v in attribute A_i is considered a DMV if $\tilde{T}_{A_i=v}$ contains a subset that represents an embedded unbiased sample (EUS) of T, where $\tilde{T}_{A_i=v} = \sigma_{T_{A_i=v}}$ [9]. The correlation between the values in $\tilde{T}_{A_i=v}$ and $\tilde{T}$ is used to discover if $\tilde{T}_{A_i=v}$ is an EUS of $\tilde{T}$, where $\tilde{T}$ is the complete set of tuples after removing the attribute A. The correlation between the values in a given table τ is computed as:

$$Corr_{v_1,v_2,\ldots,v_n} = \frac{P_\tau(v_1,v_2,\ldots,v_n)}{\prod_{i=1}^{n} P_\tau(v_i)}, \qquad (1)$$

where $P_\tau(v_1,v_2,\ldots,v_n)$ is the ratio between the number of tuples that contains the values $v_1,v_2,\ldots,v_n$ in τ and the size of τ. After computing the correlation between the values that belongs to $\tilde{T}_{A_i=v}$ with respect to $\tilde{T}_{A_i=v}$ and $\tilde{T}$, a score value that measures how good the sample $\tilde{T}_{A_i=v}$ is with respect to $\tilde{T}$ is computed as follows:

$$S = \sum_{\boldsymbol{v}\in\tilde{T}_{A_i=v}} \frac{P_{\tilde{T}}(\boldsymbol{v})}{1+|corr_{\tilde{T}}(\boldsymbol{v}) - corr_{\tilde{T}_{A_i=v}}(\boldsymbol{v})|}, \qquad (2)$$

where $\boldsymbol{v} = (v_1,v_2,\ldots,v_l)$.

The above method is computationally expensive. For example, it uses the pairwise correlation between every two attributes, which has a quadratic time complexity with respect to the number of attributes. For example, the method requires about 200 seconds to report the DMVs from a small table of 8 attributes and 768 records. We implemented a faster version of the DiMaC that reduces the running time from 200 seconds to 0.105 seconds. We give below more details about the improvements we introduced.

Fast DiMaC: To reduce the high computational cost of DiMaC, we use an index to expedite the computation of equations 1 and 2. This index contains the set of distinct values together with the subtable that is produced by selecting the records that include the value in the original table. This index has a linear space complexity with respect to the number of attributes in each table and significantly reduces the running time of DiMaC. We also remove the attributes that contain only distinct values; that is, if the number of distinct values in an attribute equals the number of tuples in the table, then that attribute is removed from the table. This is because in such data set, each set of values will appear together in a single record, which will result in having $P_\tau(v_1,v_2,\ldots,v_n) = \frac{1}{\tau}$ regardless of the table entries. The score computed in 2 will show that the most frequent value will always be the reported DMV. We also remove the attributes with single distinct values.

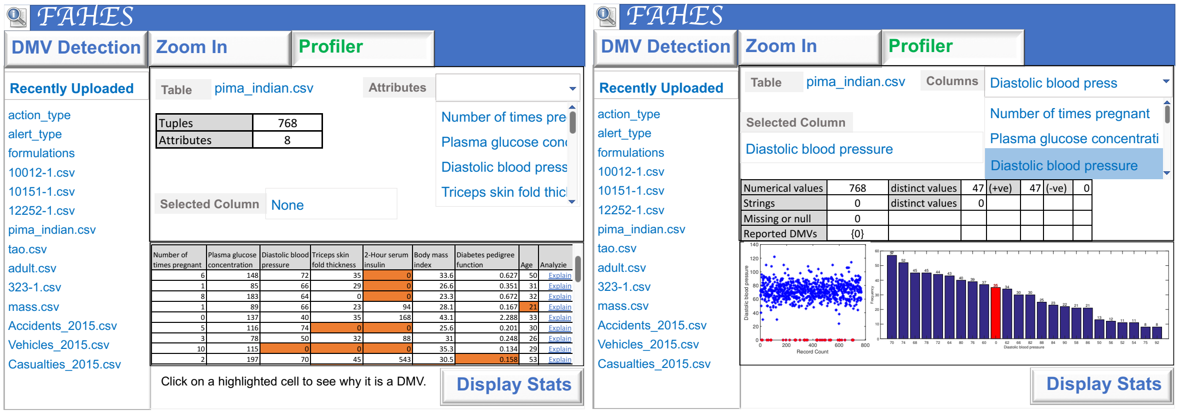
Please note that DiMaC is able to detect a set of DMVs that cannot be detected by the set of rules or by the outlier detection. The DMVs reported by DiMaC were also difficult to discover manually. However, DiMaC also reports some false positive DMVs. Reducing the number of false positives is a work-in-progress.

Table II compares the running time of Fast DiMaC with DiMaC on datasets with different numbers of tuples and attributes. The datasets were extracted from public and private repositories. Each experiment is run for a maximum of one hour and report (+1) hour if it does not terminate. The running time depends on the number of tuples, attributes and frequent values. We see that Fast DiMaC is at least three orders of magnitude faster than DiMaC depending on the size of the table.

Table II
RUNNING TIME (SEC) FOR DIMAC AND FAST DIMAC

Data set	**Rows**	**Columns**	**DiMaC**	**F-DiMaC**
Pima Indians Diabetes [7]	768	8	213	0.039
Adult [7]	32561	15	(+1) hour	11.070
DOE H. School Perf. [5]	438	18	694	0.297
SFO Museum Exhib. [5]	1242	16	(+1) hour	0.767
Avg. Daily Traf. Counts [5]	1279	9	42.3	0.197
Website Analytics [5]	3367	10	(+1) hour	0.638

Logging. We log various information about the reported DMVs, which component reported them, and the running



Tuples	768
Attributes	8

Number of times pregnant	Plasma glucose concentration	Diastolic blood pressure	Triceps skin fold thickness	2-Hour serum insulin	Body mass index	Diabetes pedigree function	Age	Analyzie
6	148	72	35	0	33.6	0.627	50	Explain
1	85	66	29	0	26.6	0.351	31	Explain
8	183	64	0	0	23.3	0.672	32	Explain
1	89	66	23	94	28.1	0.167	21	Explain
0	137	40	35	168	43.1	2.288	33	Explain
5	116	74	0	0	25.6	0.201	30	Explain
3	78	50	32	88	31	0.248	26	Explain
10	115	0	0	0	35.3	0.134	29	Explain
2	197	70	45	543	30.5	0.158	53	Explain

Click on a highlighted cell to see why it is a DMV.

(a)

Numerical values	768	distinct values	47	(+ve)	47	(-ve)	0
Strings	0	distinct values	0				
Missing or null	0						
Reported DMVs	{0}						

(b)

Figure 3. Displaying profile statistics in FAHES.

status of each component. For DMVs reported by the rule-based detector, we also store information about the rule that was violated. For example, if a value contains a repeated pattern while the rest of the values do not, then we store the repeated pattern and general information about repeated pattern violation. For each component, we store the fraction of the running time required by each component and the set of DMVs it has reported.

C. *Aggregating and Storing the Results*

Since a DMV could be detected by more than one detection component, they are consolidated to report each DMV only once. A separate file is used to store the set of reported DMVs to provide the user with an easier way to access the DMVs. A DMV that is detected by any of the detection components is considered a DMV. The system allows the user to display each table in a separate tab with all reported DMVs highlighted to make it easier for the user to confirm if the reported DMVs are correct or false positives.

III. Demonstration Plan

During the demo, we use data sets which are available online from data.gov, UCI machine learning repository, data.gov.uk, and from mass.gov. We will also show anonymized private data sets from MIT data warehouse and local companies in Qatar. The audience is also encouraged to bring their own data and test it using our system.

Figure 2 shows the available functionalities and how the user interacts with our system. Initially, the user can select a data set from the available data sets that are shown on the left panel of the figure or upload new data sets (the user can select as many data sets as she wishes). After running the system, the results will be shown in a tabular fashion that includes the name of the data repository, the table name, the column name, the DMV and the frequency for this DMV.

The user can then go to the “Zoom in” tab (Figure 2 (b)) to see the statistics about the different components of the system. These includes for each component, its running time spent and the number of DMVs being detected. When the ground truth for the DMVs in the data set are available, we will feed to the system in order to compute the precision and the recall of each component.

Figure 3 shows the functionalities that will be provided for the user to display statistics about the data. In particular, the user will see the content of the table where the cells that contain DMVs are highlighted. For columns, the user can display a histogram for the top-k frequents values where the bars of the reported DMVs will be colored in red. In case of columns that contains numerical values, the user can display the values on a 2D plot where the x-axis will represent the record (tuple) count and the y-axis will represent the value.

References

[1] R. Pearson, “The problem of disguised missing data,” *SIGKDD Explor. Newsl.*, vol. 8, pp. 83–92, 2006.

[2] J. Luengo, S. García, and F. Herrera, “On the choice of the best imputation methods for missing values considering three groups of classification methods,” *Knowl. Inf. Syst.*, vol. 32, pp. 77–108, 2011.

[3] A. Qahtan, X. Zhang, and S. Wang, “Efficient estimation of dynamic density functions with an application to outlier detection,” in *CIKM*, 2012.

[4] M. Hua and J. Pei, “DiMaC: A disguised missing data cleaning tool,” in *KDD*, 2008.

[5] data.gov, “Data retrieved from data.gov,” 2017. [Online]. Available: https://www.data.gov/

[6] M. Magnusson, “Discovering hidden time patterns in behavior: T-patterns and their detection,” *Behavior Research Methods, Instruments, & Computers*, vol. 32, pp. 93–110, 2000.

[7] M. Lichman, “UCI machine learning repository,” 2013. [Online]. Available: https://archive.ics.uci.edu/ml/index.php

[8] R. Little and D. Rubin, *Statistical Analysis with Missing Data, 2nd Edition*. WILEY, 2002.

[9] M. Hua and J. Pei, “Cleaning disguised missing values: A heuristic approach,” in *KDD*, 2007.